Team 4

Daniel Barnes - Documentation
Arie Jian - Code

Change Request

FEMR-158: don't require the user editing a user to fill out the "Change User Password" input fields.

If an administrator tries to edit a user, the "Change User Password" input fields are required and a notification gets sent back that the "password field is empty".
Action should only be taken if these fields are filled out - they shouldn't be required to be filled out.

Concept Location

Step	Description	Rationale
1	Search source code for "Change User Password"	Trying to find input field
2	Examine single result app\femr\ui\views\admin\users\edit.scala.html	Understand “edit user” interface
3	Examine femr.ui.controllers.admin UsersController.editPost()	Referenced in form submission of edit.scala.html
4	Examine femr.business.services.system UserService.updateUser()	Called by editPost() to update user
5	Search source code for "password field is empty"	Look for validation logic
6	Examine single result femr.ui.models.admin.users EditViewModel.validate()	Find check for blank password
7	Bookmark lines 51-52 which check for blank password	For later removal

Impact Analysis

Step	Description	Rationale
1	Check usage of newPassword by editPost() and updateUser()	newPassword will be null if field is left blank
2	Confirm later uses of newPassword check for null	Ensure newPassword is not dereferenced if it is null

Actualization

Step	Description	Rationale
1	Remove lines in TestDataLayerModule.java referring to non-existant IDiagnosis and IMedicationAdministration classes	Errors in pre-existing code prevent successful build
2	Remove the bookmarked lines in EditViewModel,java (Concept location: step 7)	Remove check for blank password
3	Undo step 2 and modify password constraints conditional instead	Only apply constraints if password is not blank

Validation

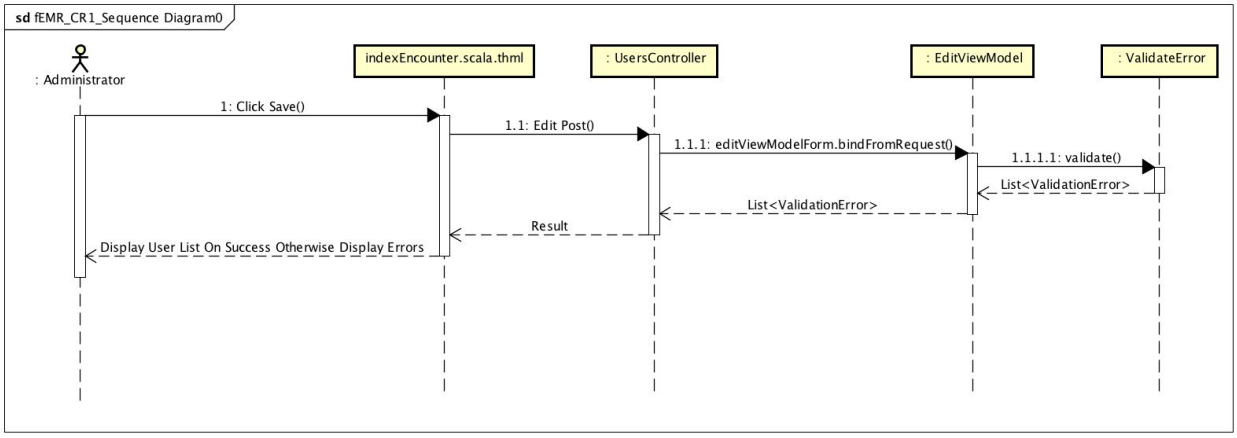
Step	Description	Rationale
	Change “about” field of user anon14 and save. Success if no error during save.	Leave password blank

Timing

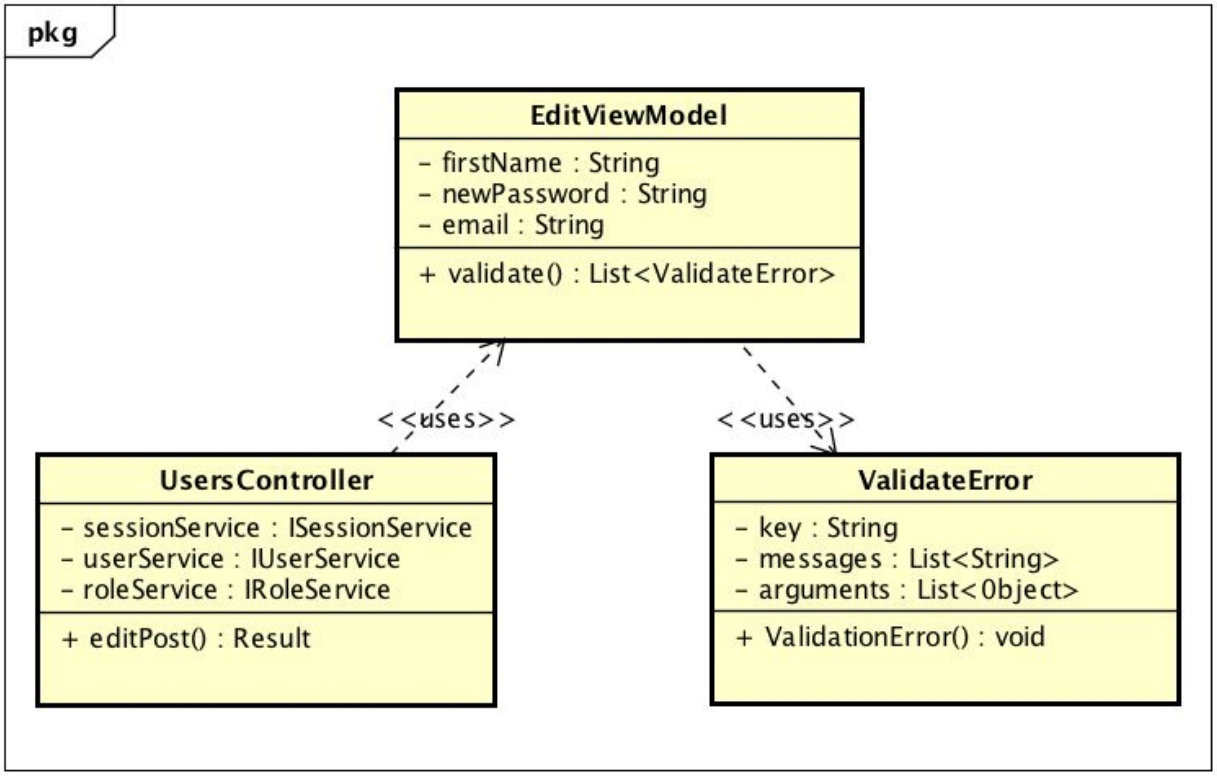
Phase	Time (minutes)
Concept Location	35
Impact Analysis	5
Actualization	20
Validation	5
Total	65

Reverse Engineering

fEMR_CR1_Sequence Diagram



fEMR_CR1_Class Diagram



Conclusions

We encountered several challenges when setting up the system to build and run fEMR. There were lines in a test procedure that reference non-existing code. We had to find and include junit libraries into the project. We initially used gitBash to do a clean compile, but it does not work properly with the PlayFramework, so we switched to using the Windows command-line. Class methods were typically large, with lots of responsibility, making them difficult to understand. The user interface did not filter choices based on the user's role, which made it difficult to understand the appropriate action to take for each role.

Concept localization was simple. A simple text string search identified where to start. The actualization was challenging because we initially determined that removing the check for blank passwords was sufficient. It was not, because the password complexity validation still rejected the blank password fields.

Classes and methods changed

- app/femr/ui/models/admin/users/EditViewModel.java
  - validate()